

**IN THE CLAIMS:**

1. (Original) A method of calling a portion of computer code in a multithreaded environment, comprising:
- receiving a call to the portion of computer code;
  - determining if the portion of computer code is currently being compiled; and
  - redirecting the call to an interpreter, if the portion of computer code is currently being compiled.
2. (Original) The method of claim 1, wherein the portion of computer code is a Java method.
3. (Original) The method of claim 1, wherein redirecting the call to an interpreter includes redirecting the call to a Java Virtual Machine Interpreter such that the portion of computer code is interpreted by the Java Virtual Machine Interpreter in response to receiving the call to the portion of computer code.
4. (Original) The method of claim 1, wherein determining if the portion of computer code is currently being compiled includes determining a setting of a flag in a control block of the portion of computer code.
5. (Original) The method of claim 1, wherein the step of redirecting the call is performed in response to a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, pointing to a JIT to Java Virtual Machine (JVM) routine.
6. (Original) The method of claim 1, further comprising:
- determining if compilation of the portion of computer code has ended; and
  - redirecting the call to a compiled version of the portion of computer code if the compilation of the portion of computer code has ended.

7. (Original) The method of claim 6, wherein redirecting the call to a compiled version of the portion of computer code is performed in response to setting a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, to point to the compiled version of the portion of computer code.

8. (Original) The method of claim 1, wherein the portion of computer code is a Java method having an associated method block and wherein the steps of determining if the portion of computer code is currently being compiled and redirecting the call are performed based on information stored in fields of the method block.

9. (Original) The method of claim 8, wherein the method block includes a field that includes a pointer that points to a Java Virtual Machine (JVM) interpreter before a Just-In-Time (JIT) compiler is loaded, points to a JIT compiler routine CompileThisMethod when the JIT compiler is loaded, and points to a routine which calls a compiled version of the method once the method is compiled by the JIT compiler.

10. (Original) The method of claim 8, wherein the method block includes a field having a pointer that points to a Just-In-Time (JIT) compiler routine CompileThisMethod when the JIT compiler is loaded and points to a compiled version of the method when compilation of the method by the JIT compiler is complete.

11. (Original) An apparatus for calling a portion of computer code in a multithreaded environment, comprising:

receiving means for receiving a call to the portion of computer code;  
first determination means for determining if the portion of computer code is currently being compiled; and  
first redirection means for redirecting the call to an interpreter, if the portion of computer code is currently being compiled.

12. (Original) The apparatus of claim 11, wherein the portion of computer code is a Java method.

13. (Original) The apparatus of claim 11, wherein the first redirection means redirects the call to a Java Virtual Machine Interpreter such that the portion of computer code is interpreted by the Java Virtual Machine Interpreter in response to receiving the call to the portion of computer code.

14. (Original) The apparatus of claim 11, wherein the first determination means determines a setting of a flag in a control block of the portion of computer code.

15. (Original) The apparatus of claim 11, wherein the first redirection means redirects the call in response to a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, pointing to a JIT to Java Virtual Machine (JVM) routine.

16. (Original) The apparatus of claim 11, further comprising:  
second determination means for determining if compilation of the portion of computer code has ended; and  
second redirection means for redirecting the call to a compiled version of the portion of computer code if the compilation of the portion of computer code has ended.

17. (Original) The apparatus of claim 16, wherein the second redirection means redirects the call to a compiled version of the portion of computer code in response to setting a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, to point to the compiled version of the portion of computer code.

18. (Original) The apparatus of claim 11, wherein the portion of computer code is a Java method having an associated method block and wherein the determination means determines if the portion of computer code is currently being compiled and the redirection means redirects the call based on information stored in fields of the method block.

19. (Original) The apparatus of claim 18, wherein the method block includes a field that includes a pointer that points to a Java Virtual Machine (JVM) interpreter before a Just-In-Time (JIT) compiler is loaded, points to a JIT compiler routine CompileThisMethod when the JIT compiler is loaded, and points to a routine which calls a compiled version of the method once the method is compiled by the JIT compiler.

20. (Original) The apparatus of claim 18, wherein the method block includes a field having a pointer that points to a Just-In-Time (JIT) compiler routine CompileThisMethod when the JIT compiler is loaded and points to a compiled version of the method when compilation of the method by the JIT compiler is complete.

21. (Original) A computer program product in a computer readable medium for calling a portion of computer code in a multithreaded environment, comprising:  
first instructions for receiving a call to the portion of computer code;  
second instructions for determining if the portion of computer code is currently being compiled; and  
third instructions for redirecting the call to an interpreter, if the portion of computer code is currently being compiled.

22. (Original) The computer program product of claim 21, wherein the portion of computer code is a Java method.

23. (Original) The computer program product of claim 21, wherein the third instructions for redirecting the call to an interpreter includes instructions for redirecting the call to a Java Virtual Machine Interpreter such that the portion of computer code is interpreted by the Java Virtual Machine Interpreter in response to receiving the call to the portion of computer code.

24. (Original) The computer program product of claim 21, wherein the second instructions for determining if the portion of computer code is currently being compiled

includes instructions for determining a setting of a flag in a control block of the portion of computer code.

25. (Original) The computer program product of claim 21, wherein the third instructions for redirecting the call to the portion of computer code are executed in response to a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, pointing to a JIT to Java Virtual Machine (JVM) routine.

26. (Original) The computer program product of claim 21, further comprising:  
fourth instructions for determining if compilation of the portion of computer code has ended; and

fifth instructions for redirecting the call to a compiled version of the portion of computer code if the compilation of the portion of computer code has ended.

27. (Original) The computer program product of claim 26, wherein the fifth instructions for redirecting the call to a compiled version of the portion of computer code are executed in response to setting a Just-In-Time (JIT) invoker field, in a control block of the portion of computer code, to point to the compiled version of the portion of computer code.

28. (Original) The computer program product of claim 21, wherein the portion of computer code is a Java method having an associated method block and wherein the second instructions and third instructions are executed based on information stored in fields of the method block.

29. (Original) The computer program product of claim 28, wherein the method block includes a field that includes a pointer that points to a Java Virtual Machine (JVM) interpreter before a Just-In-Time (JIT) compiler is loaded, points to a JIT compiler routine CompileThisMethod when the JIT compiler is loaded, and points to a routine which calls a compiled version of the method once the method is compiled by the JIT compiler.

a 30. (Original) The computer program product of claim 28, wherein the method block includes a field having a pointer that points to a Just-In-Time (JIT) compiler routine CompileThisMethod when the JIT compiler is loaded and points to a compiled version of the method when compilation of the method by the JIT compiler is complete.

---